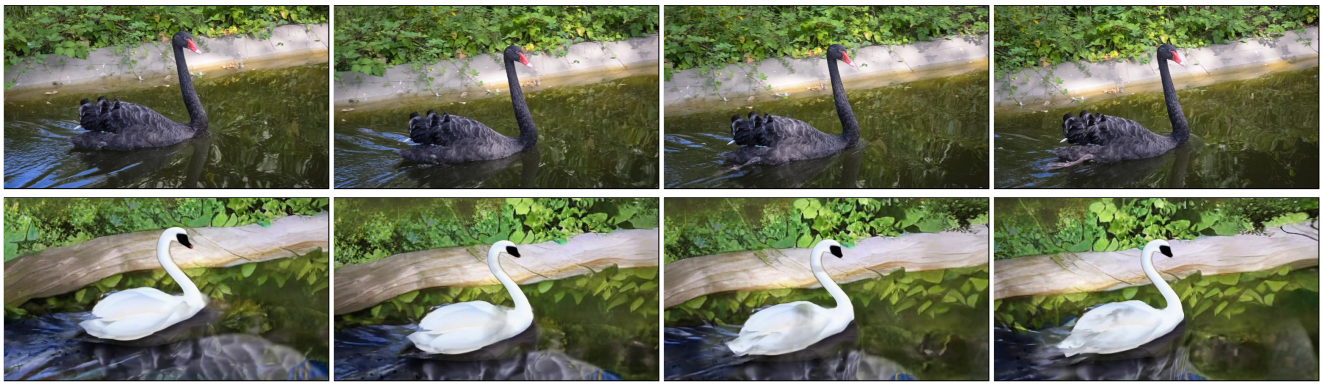
Shape-aware Text-driven Layered Video Editing
Supplementary Material

Yao-Chih Lee Ji-Ze Genevieve Jang Yi-Ting Chen Elizabeth Qiu Jia-Bin Huang
University of Maryland, College Park
https://text-video-edit.github.io

In this supplementary document, we present additional visual results (Figure 1), quantitative human evaluation (Section 1), visual comparison with the modified Text2LIVE [1] (Section 2), discussion of atlas editing (Section 3), and further implementation detail (Section 4) to complement the main paper.



"black swan → white swan" + "river → cartoon-style river"

"surfing → sail boat" + "sea → cartoon-style sea with strong wave"

Figure 1. **Additional visual results with foreground and background editing.**

1. Quantitative human evaluation

We conduct a human perceptual evaluation to compare our method with the baseline approaches and Text2LIVE [1]. We follow the Two-alternative Forced Choice protocol as in [1,4,5]. For each question in the survey, we show an input video and an editing text prompt with two resulting videos from our method and another approach. The participants are asked to select

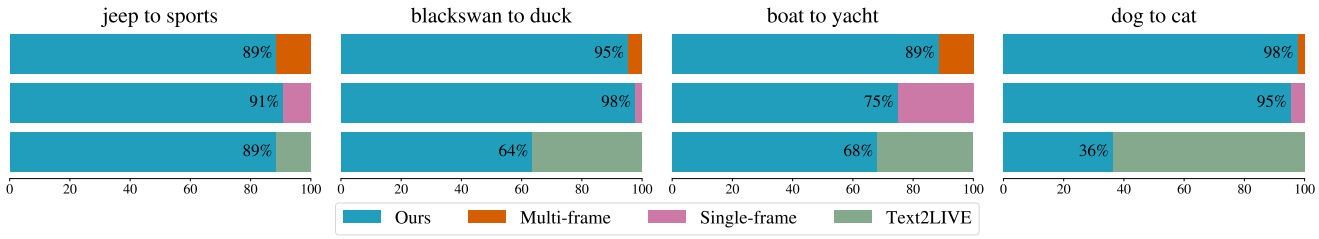


Figure 2. **Quantitative comparison by human evaluation.** We show the users' judgment of our method with each of compared baseline methods and Text2LIVE [1]. In general, most participants prefer our method to other approaches.

a video that is of higher quality and a better match to the target prompt. We collect 44 users' judgments on the comparisons of four videos, which correspond to the examples in Figure 1, 2, and 5 in the main paper.

The evaluation results of each example video are shown in Figure 2. In all example videos, most users prefer our edited videos in contrast to the results of the multi-frame baseline, as it produces temporal flickering in the video. The participants also tend to choose our method rather than the single-frame baseline with severe distortion in most cases. While comparing with Text2LIVE [1], our method generally outperforms in three videos but except the "`dog→cat`" case. This case is more challenging due to the occlusion of the foreground object. Therefore, the occlusion affects the foreground deformation to yield artifacts on the target object. As a result, some users prefer Text2LIVE's result as it does not deform the object to avoid artifacts. However, again, Text2LIVE utilizes the fixed UV for editing so that the edits' shapes cannot match the target prompt. Overall, our method is the most desirable for users with both shape and appearance editing.

2. Visual comparison with modified Text2LIVE [1]

In this section, we attempt to modify the official Text2LIVE to enable shape-aware editing. We compare the modified versions with the original Text2LIVE and our method in Figure 3.

2.1. Text2LIVE without structure loss.

Text2LIVE adopts a structure loss to preserve the original shape. Therefore, we remove the loss to see if CLIP [6] could guide the shape changes for the target prompt. Nevertheless, the modified Text2LIVE is still limited by the fixed UV mapping and thus results in the shape of the source object.

2.2. Text2LIVE with trainable UV maps.

To unlock the fixed UV in Text2LIVE, we attempt to jointly train the per-frame UV maps during the optimization. During the training, the original UV map is added with a learnable residual map for refining the UV sampling. However, the training of UV maps is not effective for changing the frames into the target shape. We found that the trainable per-frame UV makes the optimization hard to converge and thus results in noisy texture. Besides, the rendering process of Text2LIVE blends the trained layer of editing effect with the original input frames. Thus, the shape of the source object has remained in the resulting frames. Furthermore, the temporal flickering are easily observed since the per-frame UV residual are trained individually. In sum, we believe that unlocking the fixed UV maps in Text2LIVE is not feasible for shape-aware editing.

3. Editing on foreground atlases.

We demonstrate the difficulties of editing foreground atlases with general image editing models. As shown in Figure 4, the source foreground atlas is an unwrapped object texture due to the 3D transformation motion in the video. Hence, the distorted texture is not natural for the pre-trained Stable Diffusion model [7] to perform promising manipulation. As a result, although temporal consistency is still achieved, the artifacts in the edited objects can be observed due to the incomplete and distorted edited atlases.

On the other hand, we also show the editing on background atlases in Figure 5. In contrast to the difficulties in editing foreground atlases, background atlases can be directly edited by a general image manipulation model. Due to the camera motions of the video, the original background atlas can be viewed as a natural panorama image for image editing. Moreover, the edited background atlases are complete for rendering since all visible pixels in the video background are edited in the atlases.

Figure 3. **Visual comparison with modified Text2LIVE [1].** We attempt to modify the official Text2LIVE to enable shape-aware editing. We remove the shape preservation loss, but the results still remain in the source shape due to the fixed UV (as in the 4th row, w/o struct.) Therefore, we further try to jointly train the per-frame UV maps during the optimization. However, the training of UV maps makes noisy and temporal flickering in the 5th row, w/ trained UV). In sum, we believe that Text2LIVE is not feasible for shape-aware editing.

4. Implementation detail

4.1. NLA [3] configuration

We follow the same setting as in NLA [3] and Text2LIVE [1]. The maximum video length is set to 70, and the video frames are in the resolution of 768×432. The original atlases are in 2000×2000. We crop the foreground atlas to keep the object only and remove most of the empty space. The cropped foreground is resized to 512×512 to preserve more detail.

4.2. Deformation formulation.

In the deformation process, we frequently use thin-plate spline (TPS) [2]. Since the estimated semantic correspondence could be noisy, we exploit TPS to smooth the dense correspondence. Besides, TPS is also used to approximate forward warping to avoid introducing holes. The TPS is generally formed by a grid of control points with a grid size of 32. For final video rendering, we again use a temporal TPS to smooth the per-frame deformed UV maps. The total deformation and rendering take 0.3 sec per frame on average.

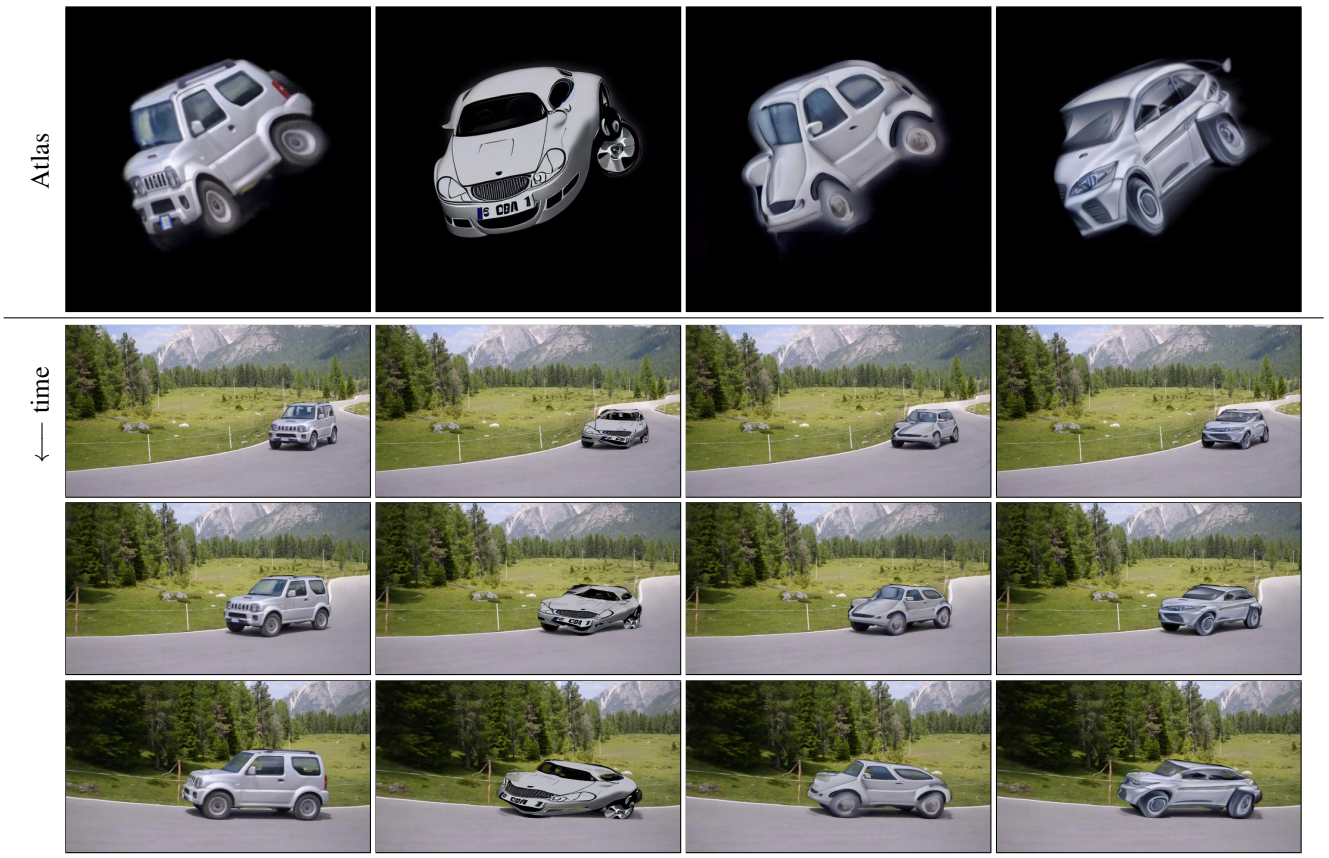


Figure 4. **Editing on foreground atlases.** The source and edited foreground atlases are demonstrated in the top row, and the resulting frames are in the same column. The foreground atlas is an unwrapped texture due to the 3D object motion in the video. The distorted atlas poses a challenge for the general pre-trained Stable Diffusion to manipulate. As a result, the artifacts are shown in the rendered frames because of the incomplete and distorted edited atlases.

Figure 5. **Editing on background atlases.** The source and edited background atlases are shown in the top row, and the resulting frames are in the same column. The background atlas can be treated as a natural panorama image for image editing. In addition, the edited background atlases can be directly used for the entire video since each visible pixel is edited in atlases.

References

[1] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. *ECCV*, 2022. 1, 2, 3

[2] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE TPAMI*, 1989. 3

[3] Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing. *ACM TOG*, 2021. 3

[4] Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. Style transfer by relaxed optimal transport and self-similarity. In *CVPR*, 2019. 1

[5] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. *NeurIPS*, 2020. 1

[6] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2

[7] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2